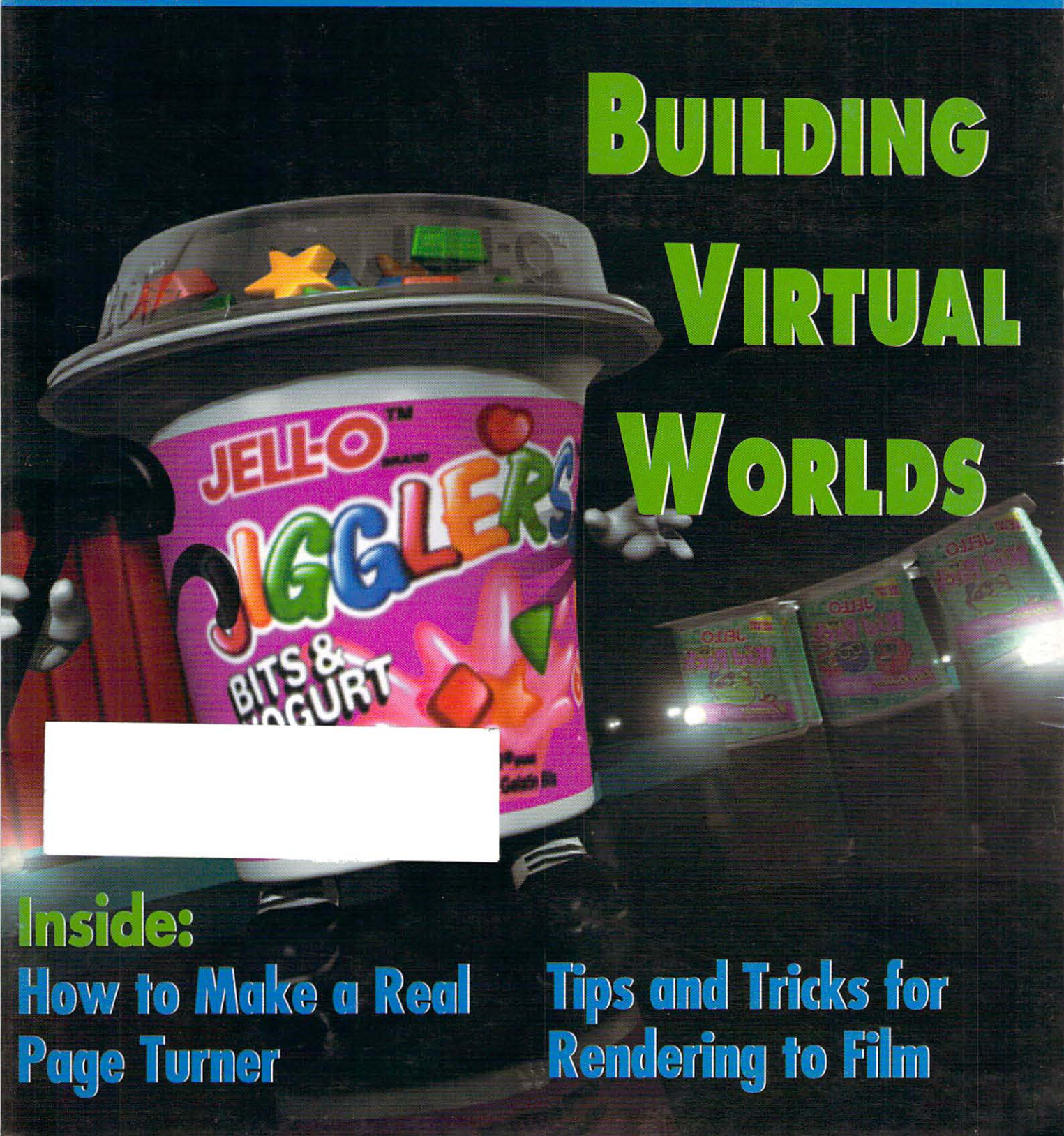
THE JOURNAL FOR LIGHTWAVE 3D® ANIMATORS

LIGHTWAVE™ PRO

a Miller Freeman, Inc. journal

Volume 4 Issue Number 7 • $8.00



BUILDING VIRTUAL WORLDS

Inside:

How to Make a Real Page Turner

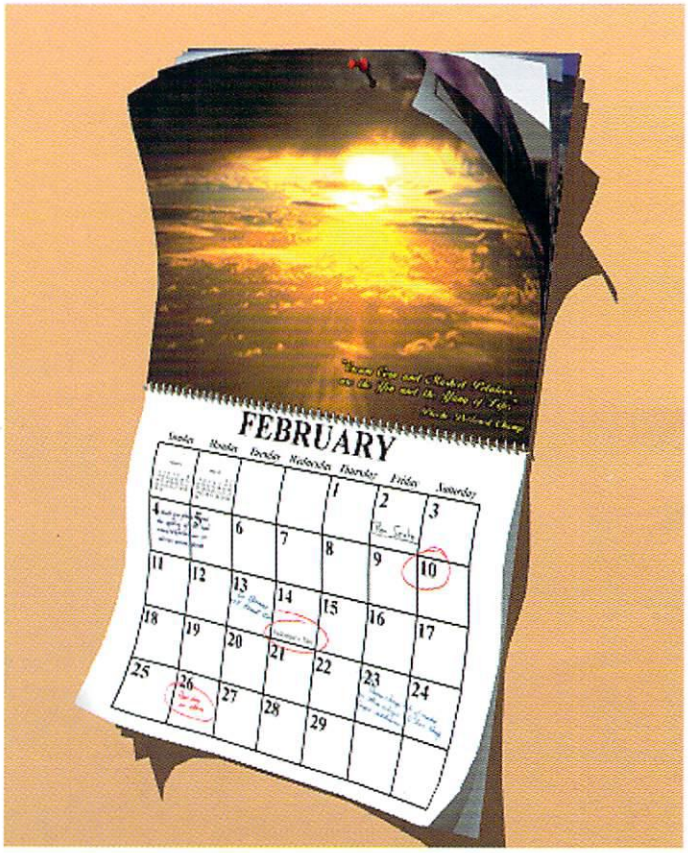
Tips and Tricks for Rendering to Film

ANIMATION STILLS

In the Cockpit

This colorful image of a helmeted pilot comes from a four-minute music video clip created in LightWave 4.0b by Dinko Cepak, who also contributed the cover image for last month's *LIGHTWAVEPRO*.

Copyright©1996 Dinko Cepak



Ah, Much Better

The flaws in this calendar are what make it so real. Notice that on the reverse side of the front picture, you can actually see the calendar page to be turned down.

Copyright©1996 Arnie Boedecker

EDITOR'S MESSAGE

by John Gross

Well, here it is, time for another SIGGRAPH already. Boy the time sure flies! By the time you will receive this issue, SIGGRAPH will have come and gone, but for me, it's still a few days away.

This year SIGGRAPH is in New Orleans, and that got me thinking about the last time I was there. It was 1989, and I was there for an NSPI convention (National Spa & Pool Institute). That was a long way away from the Computer Graphics world, and where I am now, but how I got to where I am may be a valuable lesson for many of you out there that are looking to break into the 3D world permanently. I'll make it brief ...

I had arranged to have a day off every week from my job as assistant manager of Valley Pools in Burnsville, Minn. On those days off from the pool store, I would intern (for free) at Alpha Video, a video dealership in Minneapolis. This was slightly before the Video Toaster had started shipping and they were just starting to get into this kind of thing. I originally was interning there so I would be able to get an "employee discount" on a Toaster when they shipped (I had already caught the Amiga bug from my good friend, Scott). I soon bought my Toaster, right when they shipped and before I knew it I was hooked on LightWave 3D.

Well, I just kept interning and doing whatever I could to help out at Alpha Video, and pretty soon, I was helping out with some graphics for a paying client. At this point, my contributions were recognized and I was actually given some money for helping out. As time went on, and I kept hanging out, they eventually realized I could be a valuable asset to the company and I was offered a job.

Good-bye to the pool company, which was a very secure and comfortable place to work, but I realized I didn't want to make a career out of it. And hello to the exciting world of computer graphics!

A year or two later, NewTek became aware of me and what I was doing at Alpha Video (It didn't hurt that I had started writing for VTU as well) and when the *seaQuest* project came up, I was asked if I wanted to come work on it.

see Editor's Message, page 17

TABLE OF CONTENTS Vol. 4, No. 7

4 From the Net
by Dan Ablan
Making the pages of a book turn is a simple but useful animation that can be put together quite easily in any version of LightWave 3D.

6 Rendering for Film
by Tom Williamson
Want to see your animations on the big screen? Here's what you should know about rendering your projects out to film.

8 The Imperfect World
by Arnie Boedecker
The debut of a new column focusing on how the clever use of imperfections—or 'dirty-ing up'—help make your animations more realistic.

16 LightWave Meets VRML
by Arnie Cachelin
3D is transforming the Internet into a galaxy of virtual worlds. Here's how to use LightWave to put the space in Cyberspace.

20 Flight School: Part Three
by James M. Curtis
You've modeled and surfaced your 3D bird. Now it's time to take off. In the concluding installment, create a fluid simulation of flight using morphing targets.

LIGHTWAVEPRO
a Miller Freeman, Inc. Journal

"We provide the most valuable information to people who use technology to create messages with impact."

Editor ..... John Gross
Executive Editor ..... Dave Madden
Managing Editor ..... Rosemary Picado
Production ..... Tarita Whittingham
Cover Design ..... Tarita Whittingham
Contributing Writers ..... Dan Ablan
..... Arnie Boedecker
..... Tom Williamson
..... Arnie Cachelin
..... James M. Curtis
Publishing Director ..... Michael D. Kornet
Editorial Office: Miller Freeman, Inc.
411 Borel Ave., San Mateo CA 94402
Telephone: 415-358-9500; Fax: 415-358-9732
E-mail John Gross at jgross@netcom.com, or Dave Madden at dmadden@mfi.com.

Are you interested in writing for *LIGHTWAVEPRO* or submitting images? If so, contact us at our office or electronically.

Miller Freeman, Inc., its employees, representatives or freelancers are not responsible for any injury or property damage resulting from the application of any information in *LIGHTWAVEPRO*. *All contents copyright © 1996 Miller Freeman, Inc.*

LIGHTWAVEPRO (Vol. 4, No. 7); (ISSN 1076-7819) is published monthly by Miller Freeman, Inc., 411 Borel Ave., San Mateo, CA 94402. A one-year subscription (12 issues) in the U.S. and its possessions is $48 (U.S.); Canada/Mexico, $60 (U.S.); overseas, $84 (U.S.). To subscribe, call toll-free 1-800-274-2127. In Colorado & outside the U.S. 1-303-678-0439. Allow up to 6 weeks for first issue to arrive.

Back issue orders & inquiries, call toll-free 1-800-444-4881; outside the U.S. 1-913-841-1631. Periodicals postage rate paid at San Francisco, CA and additional mailing offices. POSTMASTER: Send address changes to *LIGHTWAVEPRO*, P.O. Box 54240, Boulder, CO 80322-4240.

About this month's cover: The Jello Jigglers cover image was created in LightWave 3D by Dan Ablan of AGA Digital Studios from a character animation piece put together by Arnie Boedecker of Imagination Enterprises. The spot was used to introduce Kraft Foods' two new products to the corporation. Ablan and Boedecker worked with Seitz Video Group to complete the project.

1-800-274-2127

COLUMN

From the Net

Don't Judge a Book by Its Cover

by Dan Ablan

The months go by, the clock turns, and all of a sudden, it's time for another "From The Net" column. Of course, this is not a problem because the Internet is fraught with excellent LightWave 3D questions that make for great tutorials. This month, I've been noticing a few people wondering how to make a page turn, among other things. Much of the discussion on the LightWave mailing list covers topics about LightWave version 6.0.(Editor's note: No! It's not shipping yet!) However there are many subjects suitable for even 3.0 LightWavers. Making a page turn in a book is a simple but useful animation that can be put together quite easily—with 5.0 or not.

Book 'em Dan-O!

One particular LightWaver on the Internet asked me how he would go about building a book, then turning a couple of pages. He told me that the standard *Open Book* object in LightWave's default objects directory was not exactly what he was looking for. He needed to create a soft cover book, similar to the LightWave Reference and User Guides. His render times were exceeding 30 minutes, even on a Pentium 90. As it turns out, he was building each page and stacking them. This means LightWave has to render every single page, and at 200 plus pages with a texture, well, you get the idea. Because of this, he never even got around to making the pages flip and turn. In reality, building the book is very simple, and making the pages within it flip is just as easy.

Begin this project by entering Modeler. Save any work you have and clean out the layers by pressing (shift-n) for new. Create a box with these settings:

	Low	High	Segments
X	0 m	2.7 m	40
Y	0 m	450 mm	1
Z	-1.75 m	1.75 m	1

Since the animation will primarily consist of a few of the book's pages turning, the book will be modeled already open. Because it's a soft cover book, an extra large box doesn't need to be modeled as would a hard cover book.

At this point, it's important to think ahead to the surfacing of the object. You have a couple of options for creating the page look on the edges. You can set a bump map, or actually scan in an image of pages. Scanning works very well, and avoids that computer look, giving a more natural appearance. However, many LightWavers don't have access to a scanner, so in this particular tutorial, a bump map will be used. Since bump maps and procedural textures are based on a linear axis (X, Y, or Z), and the pages need to have a slight curve, a normal bump map won't really work. A bump map on a curved object won't curve. It will be applied straight through the object. So, the first thing to do is to set the surfaces and save this object.

Surfaces

This object is one side of the book opened, which will be made to look like a clump of pages. That means you need at least three surface settings. The bottom won't be seen in the animation, but you can set the surface if you like.

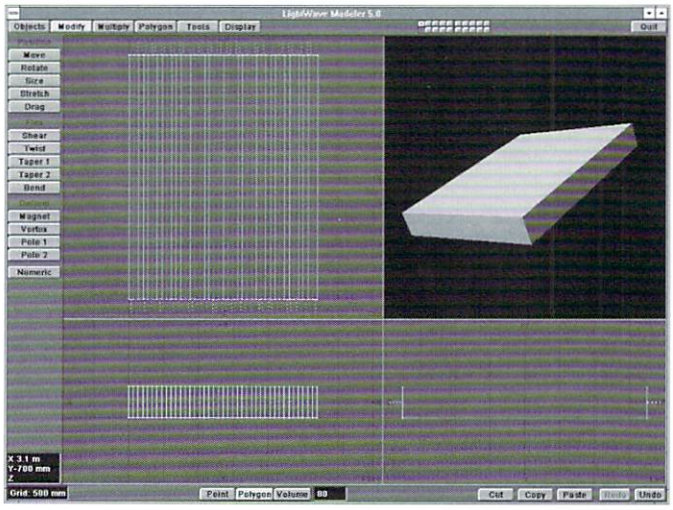


Figure 1

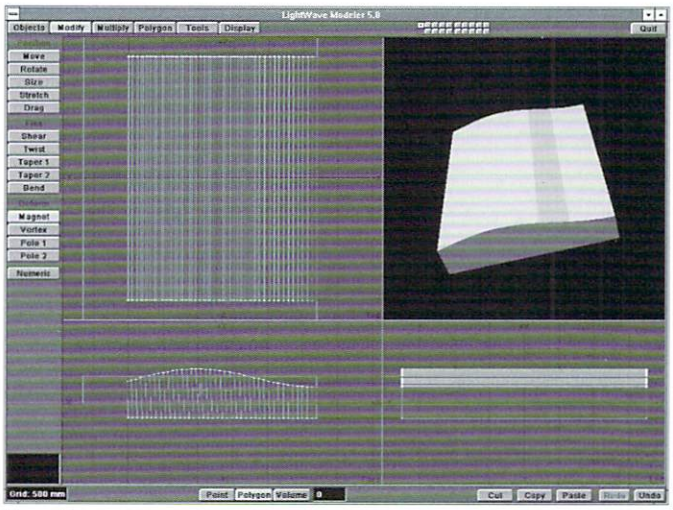


Figure 2

- First, select the polygons on the positive Z, and negative Z ends of the object as in Figure 1.
- Name these polygons Pages_Z. Do the same for the polygons on the X and the polygons on the Y. Mirror the object on the 0 X axis, and save this object as BOOK_1.LWO.
- Now you need to shape this object into one side of the book. Using the

Magnet tool from the Modify menu, draw out around the top front points in the Face view. With the right mouse button, drag up and over to the left slightly, creating a hump in the pages as in Figure 2.

•Once the hump is to your liking, use the Magnet tool again around the bottom points in the face view to give a hump to the bottom of the pages, but not as severe as the top.

•Next, select the far left points of the box from the Face view (totally 4 points), and use the Set Val command from the Tools Menu to move them equally to 0 X.

•Now, press (CTRL-t) for the Drag tool and drag the bottom right points (from the Face view) out to the right. Doing this will create the angled edge of the book.

•Lastly, drag the five left top points down to create a stronger curve near the center of the book. Your final curved book page should look like figure 3. Save the object as BOOK_2.LWO.

As the Page Turns

For the pages, all you need to do for modeling is replicate the top of the page clump.

•Using the Point selection tool, Lasso around the right side of the book's top points, as in Figure 4.

•Copy these points and paste them into a new layer. If the back points copied along with the front, delete them, you only need one row of points.

•In order, select the row of points and press (CTRL-p) to make an open curve.

Next, extrude the curve to the length 3.5 m, and enter 40 for the Segments. Click OK and press Enter. Now you have a page ready to be animated. Give it a surface name and save the object as Page_1.LWO. You may want to surface and save the object again so that you can place a different image on each page.

An Easy Set Up

Now that your objects are built, load them into Layout. From the Objects panel, select Book_1.lwo as the Current Object. Select BOOK_2.LWO as the Metamorph Target. Slide the Metamorph Amount to 100%, then change the Current Object to BOOK_2.LWO and change its Object Dissolve value to 100%. Next, from the Surfaces panel, enter the following values for the PAGE_Z surface:

Surface Color	180, 180, 180
Luminosity	0%
Diffuse	100%
Specularity	10%
Glossiness	Low
Edge Transparency	Normal
Smoothing	On
Max Smoothing Angle	40.0
Bump Map	Fractal Bumps
Texture Opacity	100%
Texture Size	10, .01, .03
Texture Amplitude	-100%
Frequencies	3

Once these settings are in place, set them to the PAGE_X surface, but change the bump map size to X .03, Y .01, and Z 10.0. Click Use Texture and return to Layout. Figure 5 shows how the bump map applies itself to the object giving the appearance of pages, while following the curve of the object because of the metamorph.

Here, you can add variations to the texture sizes as you like. The values listed here are only one way of setting the surface.

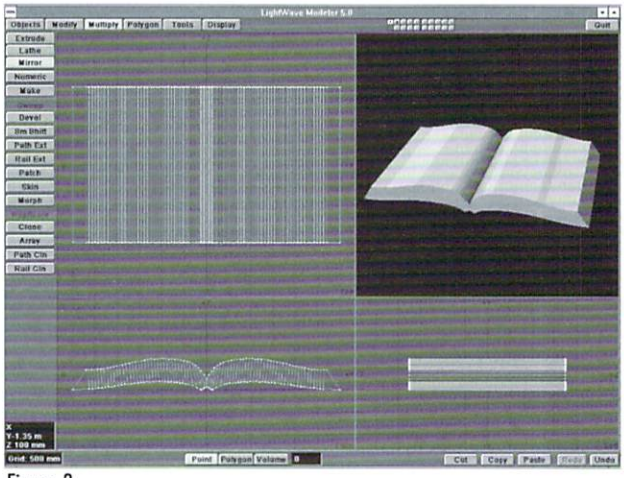


Figure 3

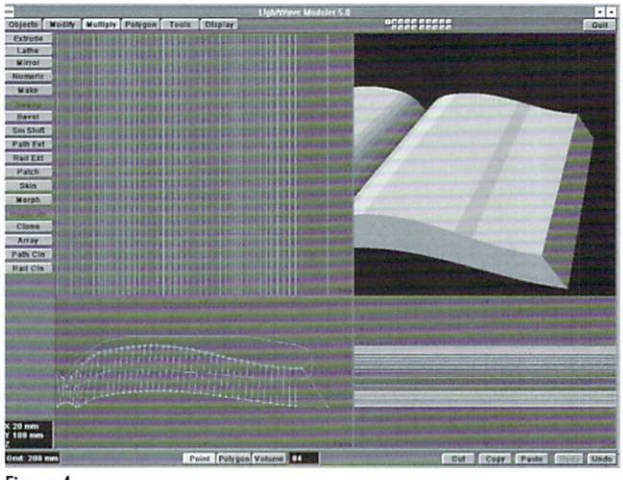


Figure 4

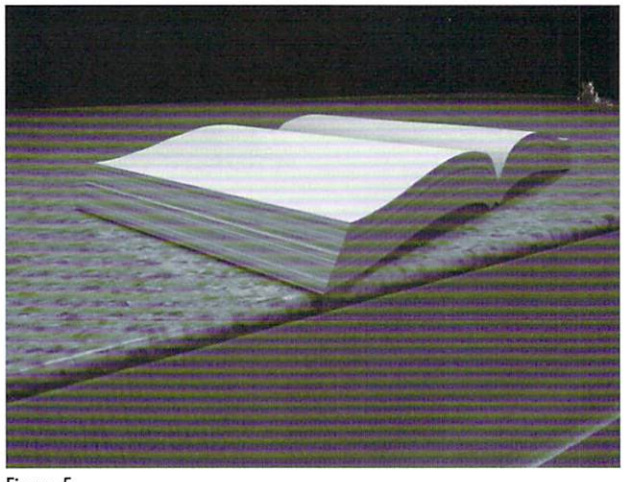


Figure 5

•Find an image, scan, or framegrab of an open book to map onto the PAGE_Y surface. Save the object once you have the settings the way you like.

•Now load the PAGE_1 object, and move it into place over the right side of the book. It's a good idea to parent the page to the book so you only have one object to worry about when putting the book into a full scene.

see From the Net, page 17

LightWave on the Big Screen

What to Remember When Rendering for Film

by Tom Williamson

The great majority of animators dream of sitting in a crowded theater, munching on popcorn and watching their own digital creations come to life on the silver screen. The reality of outputting your animations on 35 mm film is not as distant as it may seem. With the following information and a few dollars (perhaps quite a few, let's be realistic) you could be well on your way to filling that big white void with stunning computer-generated magic.

The Terminology

The first thing you have to know when you're going to output to film is a few of the terms from the motion picture industry. You need to sound like you know what you're talking about if you want people to take you seriously. We'll discuss only 35 mm film in this article, there are several film formats and each has it's own lingo, 35 mm is by far the most popular and accepted.

First off, let's discuss the playback differences of film and video. Normal film playback is at 24 fps (frames per second) as opposed to video's 30 fps. This in itself is enough to drive you nuts. For the proper playback of film on video the film must be "frame expanded" to 30 fps. This process is called 3-2 pulldown. In 3-2 pulldown every fourth frame is printed twice, hence the 3 and 2. If you do the math you'll see that by adding a frame after every fourth frame, the original 24 frames, they will add up to 30 (Are you completely confused yet? It gets worse!).

To mask the fact that every fourth frame is on screen twice (and tends to look jumpy when played back), a process of field integration is used. Every frame of video has two fields, alternating horizontal bands of image data that interlace. Each frame of video is 1/30th of a second, each field is 1/60th. This makes for smooth playback but sacrifices resolution. The 3-2 pulldown process can take the odd and even fields of two adjoining frames and create an "in-between" frame. This is done to the double printed frames and results in proper 24 fps to 30 fps playback that's very smooth. To see this process at work grab your copy of the THX wide-screen edition of *Star Wars* (you do have it, don't you?) and pop it in your VCR. If you have a steady still-frame you can frame advance through a scene and see the "in-between" frames at work (they'll be the flickering frames).

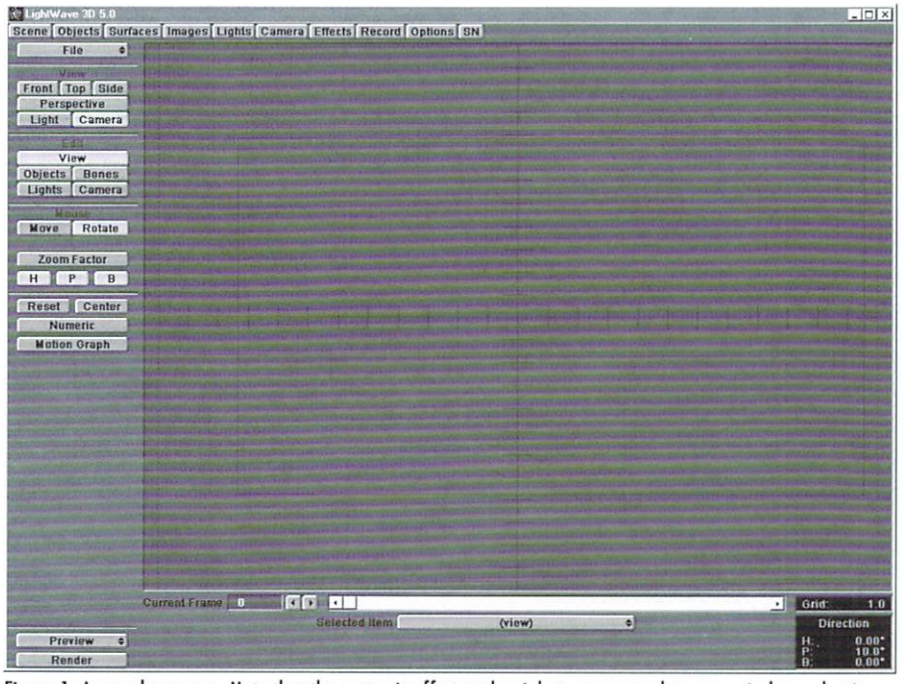


Figure 1: An academy crop. Note that the center is offset to the right to accommodate an optical sound strip.

Another mind-muncher is film measurement. Film is measured by feet plus frames. There are 16 frames per linear foot of film. If I have an image sequence that's 39 frames long it would be 2 feet plus 7 frames, which is usually displayed as <2+7>. Every piece of film has key numbers at the bottom of the strip. These number progressively count the feet. If you look at the film you will see a four digit number with a bullet "Σ" every 16 frames. Frames with a bullet beneath them are foot markers. The next frames would be 1, 2 and so on to 15. Then the process starts over with the next foot marker. It should also be noted that there are four perforations ("perfs") per frame and there are tiny key numbers that relate to these. These numbers are rarely used and are best ignored.

Next is the issue of aperture. Normally film is shot at full aperture (the full frame is exposed) and projected at *academy* (cropped to a ratio of 1:85). This is very important to know when you're rendering to film. The academy crop also is off center to the right of the frame, which allows for the optical sound stripe. To get your framing right use this trick: Go to the camera panel and set your resolution to 2048 x 1556 (normal film res or "2K") and use square pixels. Now go back to Layout and turn on the field chart. Note the hash marks, we'll use them to set up our academy crop. Hold down the "l" key (to activate limited region) and pull in the left side 2 3/4 hash marks. Pull the top down and the bottom up 4 1/5, and pull the right half way to the first mark. This is the normal visible area of projected 35 mm film (Figure 1). If you leave this limited region on while you animate, and keep your composition in the box (keeping in mind what will be the new center of frame), you should see predictable results on the screen.

For render tests I always crop the frame so no one is surprised later when they aren't seeing the

full image (besides, things just look cooler in letterbox, right?). It is, however, very important to render your final images at full aperture (filling the frame) because the academy crop is more of a guideline than a rigid standard and sometimes more or less of the frame is visible. Also, when the film is later transferred to video more of the top and bottom (and, unfortunately, less of the left and right) will be visible. For a more in depth look into aspect ratios and apertures see John Parenteau's article "Digital Cinematography" in the May 1995 issue of LIGHTWAVEPRO.

Rendering Tips

The real trick to getting computer animation to look good on film is to render it to look as if it was shot by a motion picture camera. This means intentionally adding all the little things that make film look like it does. Here's a few of those characteristics:

•**Motion Blur–** This is very important. With the exception of very high speed photography, motion blur is almost always present. To get your motion blur to look right first get the blur length to a realistic setting. This can be difficult to determine because in photography the length of the blur is determined not only by the speed at which an object is moving but also the camera's exposure (how long the film was exposed to light). It's best to experiment and get a length that looks esthetically pleasing but falls into the realm of reality. After you determine your length go back and determine what settings of anti-aliasing and dithering will be necessary to make the blur look realistic.

•**Film Grain–** This is a characteristic of film that is continually battled with by classic cinematographers. Seems strange that we would want to intentionally add it to our images, but it's very important to achieving a film look. To add realistic film grain I highly recommend picking up Steven Worley's Filmgrain plug-in (Figure 2) which will be available soon from NewTek. If nothing else turn on animated dither with 4x dither. Although some grain will naturally be introduced when you record out to film, it's worth it to render it into the image. The final look is more natural.

•**Light and Contrast–** To keep your images looking more cinematic try to limit the use of ambient light, and illuminate your scenes in pools of light. Also try to break up the light on your objects by using projected images and shadows. Keep in mind that the images on your computer and video will look different on film. The blacks will be richer and the whites will be brighter. On a video screen you can't have true black. This is because the screen is illuminated. Just look at the screen when the monitor's turned off. It's gray! On film the blacks are created by blocking the projector's light, resulting in true blacks (depending on the light level in the theater). The whites on film are just transparent plastic, so if the screen is kicking back a lot of light (and they usually do) you should have bright, clean whites.

•**Strobing–** This is the stuttering effect of viewing only 24 frames a second. Although you should not try to make your animations unpleasant to watch, this phenomenon is important. In the video world field rendering is used to get around strobing, but there are no fields on film. Luckily you don't really have to do anything to introduce strobing, it should occur naturally. Just fight the urge to field render and you should be okay.

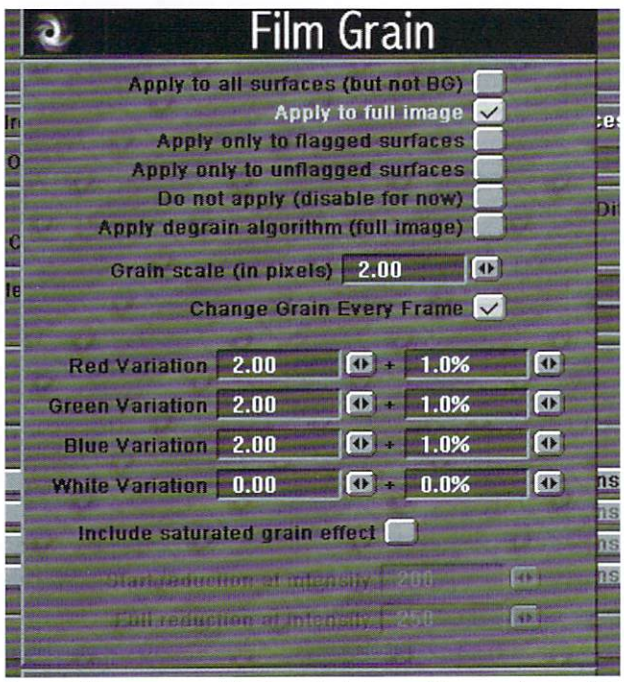

Film Grain
Apply to all surfaces (but not BG)
Apply to full image
Apply only to flagged surfaces
Apply only to unflagged surfaces
Do not apply (disable for now)
Apply degrain algorithm (full image)
Grain scale (in pixels) 2.00
Change Grain Every Frame
Red Variation 2.00 + 1.0%
Green Variation 2.00 + 1.0%
Blue Variation 2.00 + 1.0%
White Variation 0.00 + 0.0%
Include saturated grain effect

Figure 2: The settings panel for Steve Worley's new Filmgrain plug-in, which will be available soon from NewTek.

•**Composition–** When you're animating for film take advantage of the 1:85 aspect ratio. Watch a few movies with great cinematography to see how shots are composed. Use the frame to it's full effect. One technique is the rule of thirds. This is an age-old photographic method that makes for great composition. It's also very easy to do. Just divide your frame with imaginary lines into thirds, left to right and top to bottom. Where these lines intersect is where you frame your subject matter. Another thing to think about is how you "photograph" your scene. Try to keep your camera moves "realistic". In other words, don't do anything with LightWave's camera that couldn't be accomplished with a real camera. Think of the tools of a film crew, like cranes and dollies, and limit yourself to the capabilities of this equipment. Don't move too fast or whip around to much. Just think in real-world terms and ask yourself how a real cinematographer would shoot this scene.

Scanning

If your animation is going to be integrated into live action footage (35 mm) you will need to have the film scanned. Scanning is the technique of capturing the film frames as electronic images. There are several digital film services out there, so pick one based on reputation and cost. Usually the original camera negative is delivered to these facilities. It is then scanned and delivered to you on Exabyte tape (I'll discuss Exabyte tapes in the next section). When you order your scan they will probably ask if you want full or academy aperture and what resolution you want. Most often you'll need full aperture 2K images. These are 2048 x 1556 at most facilities, but check with the service before scanning begins. If you plan to move into the image (like a zoom) you may want to go to 4K, to prevent pixelation (where individual pixels become visible). Be sure to check the footage when it arrives for accuracy. Using a small strip of the original film (a "match clip"), check the framing and color. The process of scanning is still in it's infancy and subject to flaws.

Recording Out

The last thing to discuss is output, or recording out to film. This is where the money comes in. This is still a pretty expensive process. Most digital film service companies charge a setup fee, a per-frame charge and the lab costs. Expect to pay from hundreds to thousands for output of even a fairly short animation.

The technique for getting your frames on film varies depending on the platform your on. The easiest way is to use an SGI (whoa, hold on, I'll get to the Amiga). Most film recording services prefer SGI format frames (.rgb) delivered on Exabyte tape in the TAR format (Don't worry I'll go over each of these). The SGI format is the standard in high-end work, and LightWave has no problem rendering .rgb with the aid of HIIP.

Unfortunately it's not a very efficient format when it comes to storage. Most film output is at 2K, and a single frame in SGI format is over 9.5 MBs! If your going to do film work you better pick up a couple of 9 GB drives. If storage is a problem, I recommend converting the frames to IFF, they'll be smaller files and more manageable, just remember to convert them back before you output them. To send off your frames to the recording service just back them up on Exabyte and send them off. Exabyte is a tape backup medium that uses 8 mm video (actually data grade) tape. On the SGI it's as simple as connecting the Exabyte drive (they're

see Rendering for Film, page 17

The Imperfect World

The Monthly Morph

by Arnie Boedecker

The world we live in is far from perfect—at least mine is. The homes we live in have flaws: a tear in the wallpaper, a stain on the rug, a scrape along the wall, spills in the kitchen. Our cars, most often, are not perfect either. There's the occasional scratch, the annoying dents, the rusted areas. Even now, as I write this, I am using a pen minus the cap with one end chewed up. So why is it that so much of 3D animation looks so perfect? C'mon, we're all guilty. Perfectly clean walls, shiny floors and gleaming automobiles. Of course, many animators do make it a point to *dirty-up* their animations, and it is their work that often gets recognition: "It looks so real!"

I figured that it was time for a column on creating some of our world's imperfections. In future articles, I hope to cover such topics as Home Flaws, The "Junker" Car, and most often, just general imperfections. This is an area that we are all learning to master. I don't claim to be an expert real-world surfacer. In fact, much of my work is as lacking in that real-world touch as the next animator. That's all the more the reason to tackle this plague head-on! From column to column, the topic may vary from modeling tutorials, to surfacing tips, to Layout functions, or (like this column) a combination of these. Feel free to contact me with question or topic suggestions.

Okay, let's move on to this month's tutorial. Recently, I created a scene where I needed a calendar hanging on the wall, in the background. Easy enough. I just created a couple of flat polygons, mapped a calendar image on it, and placed it on my wall. A quick test-render let me know that I was falling into the same old trap again. What we have here is a need for imperfection!

When a calendar is stuck to the wall with a tack, it takes only a matter of moments for its potentially perfect flat structure to distort. The weight of the paper makes the corners sag, and humidity causes the pages to warp. Let's focus on these imperfections to make our calendar look a bit more realistic.

One Ringey Dingey

I decided to go with a spiral-type calendar, rather than the stapled-type, purely for detail purposes. With a spiral holding the pages together, the calendar is slightly more interesting to look at (I need all the help I can get!).

To make the spiral, first create a Disc, on the Z-axis, with 16 sides, leaving the center at the origin, and radii values of (0.75mm, 0.75mm, 0). Set both the Top and Bottom Values to zero to make the disc flat.

Next, use the Move Tool (Modify Menu), to move the disc up 2.75mm on the Y-axis. To make the spiral from this disc, use the Lathe Tool (Multiply Menu) with the following settings:

Start Angle:	0
Ending Angle:	18,000
Sides:	800
Axis:	X
Center:	0, 0, 0
Offset:	340mm

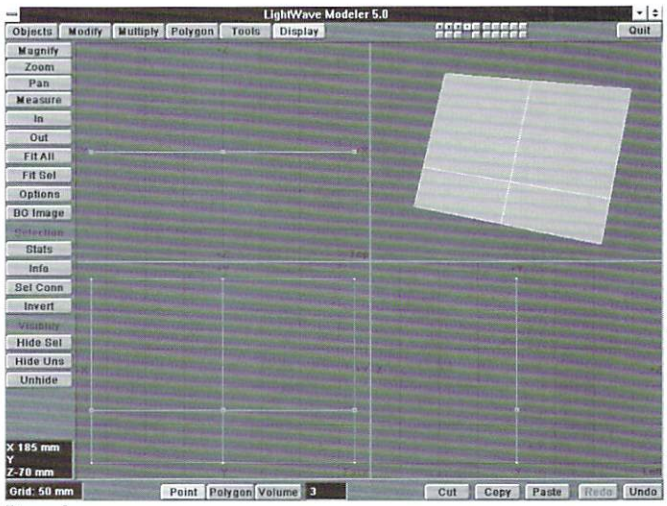


Figure 1

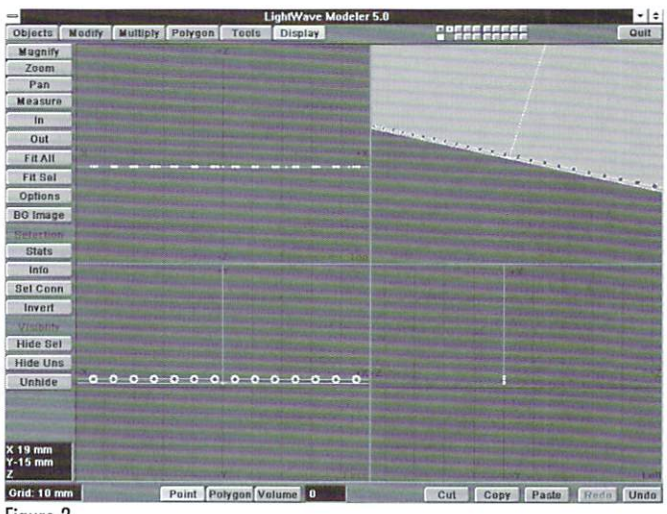


Figure 2

Once you have your spiral, zoom in close to a section of the spiral (using the period key) and select a few polygons to make sure that they are facing outward. If they're not, hit the F-key to flip the polygons.

Next, use the Center Macro to center the spiral around the origin. Then you can use the Rotate Tool (Modify Menu) to rotate the spiral 90 degrees on the X-axis to make the spiral ends a bit more visible. Then move the spiral -1 mm on the X-axis to center it up again. Press the Q-key, and assign the spiral a surface name of Calendar Ring, and save the object as CalRing.lwo.

Before we move on to the calendar pages, let's do one more thing. The spiral we just created is what holds the calendar pages together, so we are going to need to cut holes into the calendar pages slightly larger than the spiral itself. What we'll do here is create a "cookie-cutter" to be used with the Drill Tool later.

Go to a new layer and make another Disc, using the same settings, only with radii values of (1.05 mm, 1.05 mm, 0). Then, using the Move Tool, enter these numeric values (-165.9 mm, 2.8 mm, 0). Zoom in on your new disc and put the spiral into a background layer. You should now see that the disc is encircling the upper left-most rung of the spiral. This defines the area later to be cut from our pages. To duplicate our "cutter" for each upper rung, use the Clone Tool (Multiply Menu), to create 49 clones, with an X-Offset of 6.8 mm. Save this object as TopCuttr.lwo. To make the cutter for the bottom rungs, just move the top cutter (-3.4 mm, -5.6 mm, 0) to fit it over the bottom rungs. Save this object as BotCuttr.lwo.

Worth a Thousand Words

Our calendar's pages will consist of two types. The top pages, pinned to the wall, are the Picture Pages. These pages will have the picture side visible, which could be anything from automobiles, landscapes, puppies and kittens, to other less tasteful things. The lower pages that hang from the spiral are the Date Pages. These pages have the monthly calendar visible. We'll get to these pages a bit later. For now, let's work on the Picture Pages.

To start out, create a box with the following settings:

Axis	Low	High	Segments
X	-170 mm	170 mm	2
Y	1.5 mm	240 mm	2
Z	0	0	1

In the Left View, grab the three center points, between the top and bottom of the box, and move them -50.75 mm on the Y-axis (Figure 1). Next, select the two polygons to the left of the Y-axis in the Face View, and hit the Z-key to delete them. Now select the top of the two remaining polygons, and Triple it (Polygon Menu). Deselect the inner polygon that is created, so that only the corner polygon is selected, and use the Subdivide Tool in Faceted Mode (Polygon Menu), and subdivide this corner a total of three times. Deselect all polygons and use the Mirror Tool (Multiply Menu) on the X-axis to create the left half of the page.

With this page in the foreground, put the TopCuttr.lwo object that we made earlier in the background. Now use the Drill Tool (Tools Menu) on the Z-axis, selecting the Tunnel Option, to cut out the needed holes in the page (Figure 2). Make sure all of the polygons are facing forward, and assign the surface name Picture.

Copy this entire object to a new layer, and hit the F-key to flip the polygons so that they are facing the rear. Assign a surface name of Picture-Back to this object, then Cut and Paste it back into the Picture Page layer.

Finally, Merge Points (Tools Menu) to get rid of the duplicate points we created. What we have just done allows us to surface the two sides of the page separately. On the back of each Picture Page is the monthly calendar that will face out when the picture is turned down, so we needed the two different surfaces. Save this object as CalPic.lwo.

Now we want to create a morph object for this perfectly flat page so it becomes *imperfect*. The Picture Pages will be tacked to the wall at the top-center, so gravity tends to pull and roll the top corners away from the wall. This is the effect we want to create with a morph.

Use the Rotate Tool to rotate our Picture object -45 degrees on the Z-axis, leaving the center at the origin. See what we did? Now, the subdivided polygons of the top right corner are in even horizontal rows. What we want to do is rotate these rows away from the wall. Select all of the subdivided right corner polygons (128 total). It is important that the initial rotation uses the exact value of the bottom edge to be rotated, otherwise smoothing problems could arise. The

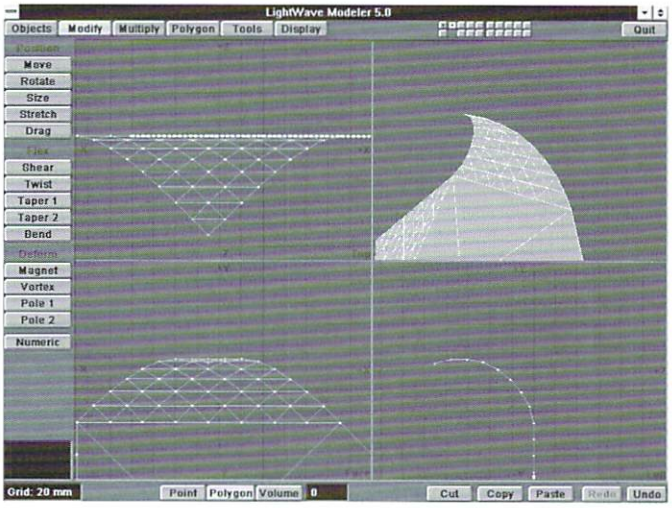


Figure 3

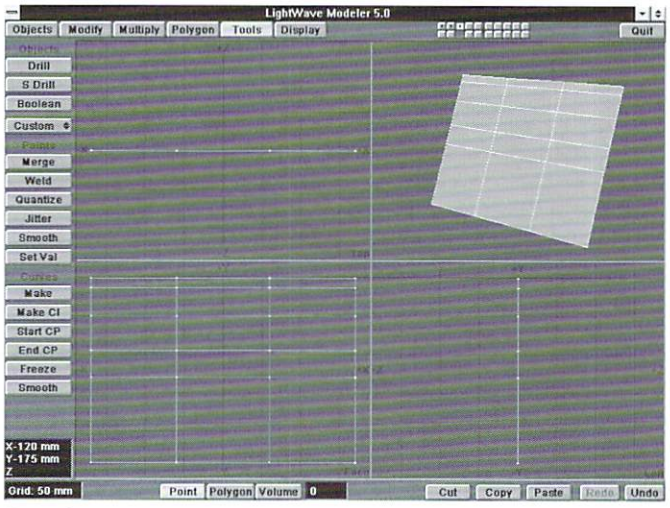


Figure 4

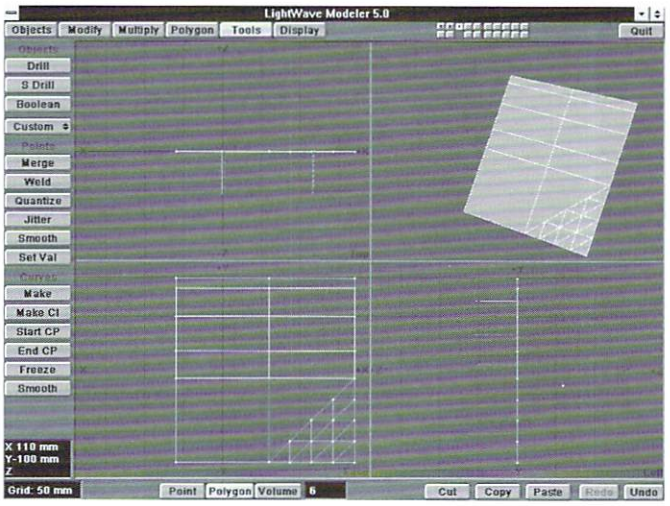


Figure 5

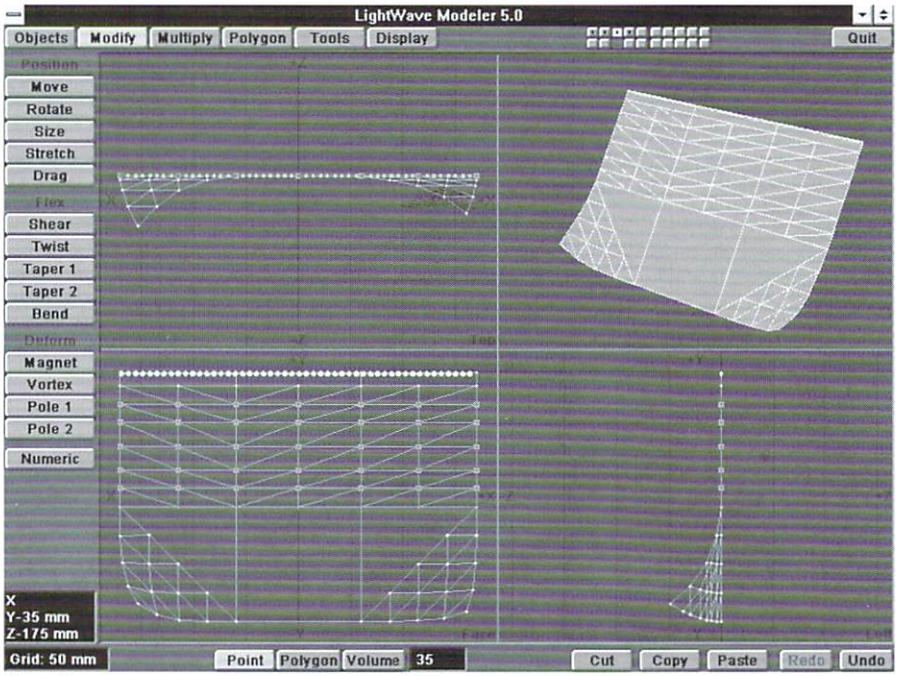


Figure 6

bottom row of points all have a Y value of 169.7056 mm, so using the Rotate Tool, hit the (N)key to bring up the numeric requester. We want to rotate on the X-axis with a center of (0, 169.7056 mm, 0). The rotation amount is up to your tastes, I used a value of 15 degrees.

From here, switch to the Left View, deselect the bottom row of polygons (You should now have 98 selected). Then zoom in on the selected polygons in the Left View. Place the cursor right at the bottom of the lowest row of selected polygons and rotate them to your tastes. Then deselect the bottom row of selected polygons again. Rotate from the bottom of selected polygons, and so on, until you reach the top (Figure 3).

When you are finished with the right corner, you need to do the same thing to the left. Rotate the page 45 degrees (center at origin) back to its original position, then rotate it another 45 degrees to put the left corners polygons into even rows. Select the left corner polygons and rotate them as we did with the right corner. The first rotation still needs to be done numerically, using the same numerical values.

When you finish rotating all of the left corner polygons, rotate the entire page back -45 degrees on the Z-axis (center at origin) to its original position. Assign the surface name, Morph, to the entire object and save this object as CalPic.mph.

What Day Is It?

We will create the Date Pages in much the same way we did the Picture Page. Being that the Date Pages hang from the spiral and are not tacked down, they will distort in a different manner, which means we need to model them differently as well.

To start, create a box with these settings:

Axis	Low	High	Segments
X	-170 mm	170 mm	3
Y	-1.5 mm	-240 mm	5
Z	0	0	1

You'll notice that, including the very top and bottom of our box, there are 6 horizontal rows of points. The top and bottom of the box will not move, but we need to shift the levels of the 4 inner rows of points.

In the Left View, select the first row of points from the top. Use the Set Value Tool (Tools Menu), and enter the value -14 mm for the Y-axis. Deselect these points, and select the next row down. Again, use the Set Value Tool with a value of -50 mm on the Y-axis. Then set the value for -95 mm for the third row, and -130 mm for the fourth row.

We also want to shift the two inner vertical rows of points. From the Top View, select the inner row to the left and set the value to -60 mm on the X-axis. Then select the inner row to the right setting their value to 60 mm on the X-axis. When all the rows are shifted as described, your box should look like the one pictured in Figure 4.

Next, grab the five polygons all the way to the left and hit the Z-key to delete the left-third of the box. Select the bottom right polygon and Triple it. Deselect the inner polygon created, so just the corner polygon is selected. Subdivide this corner polygon two times, and then deselect all polygons.

Next, from the Left View, skipping the top row, grab the next three center rows of polygons for six polygons total (Figure 5). Triple these polygons and subdivide once. Then, to create the left-third of the page, select all of the right-third polygons (42 total) and mirror them over the X-axis.

Now, put the BotCuttr.lwo that we made earlier in a background layer and again use the Drill:Tunnel Tool as we did with the Picture Page to cut out the needed holes. To surface both sides of the Date Page as we did the Picture Page, make sure all of the polygons are facing forward, and assign the surface name, CalendarDate. Copy the entire object to a new layer, flip the polygons and assign the surface name, CalendarDate-Back. Cut and Paste this object back into the Date Page layer, and Merge Points. Save this object as CalDate.lwo.

To create the morph object for the Date Page, do the bottom corners as we did the top corners of the Picture Page. Rotate the object 45 degrees on the Z-axis (center at origin) to do the right corner. Grab all of the right corner polygons (32 total), and rotate to your tastes. The initial rotation, as before, needs to be done numerically. The top row of points have a Y value of -212.132 mm. When done with the right corner, you'll need to rotate the object -45 degrees twice, as before, to do the left corner. When finished with the left corner, you'll need to rotate the object 45 degrees back to it's original position.

Since the Date Pages are not tacked down, humidity will sometimes make the pages warp out slightly.

To do this, go to the Left View, and zoom in on the Date Page. From the top of the page, grab rows of points three through seven, for a total of 35 points selected (Figure 6). Use the Magnet Tool (Modify Menu) on the X-axis, with these settings to warp out the Date Page:

Axis	Radii	Center	Offset
X	0	0	0
Y	65 mm	-65 mm	0
Z	120 mm	-32 mm	-10 mm

Assign a surface name, Morph, to the entire object and save the object as CalDate.mph.

Tack It Up

Time to put up our calendar. The first thing to mention is, if you want a number of calendar pages visible but with different surfaces, you will need to load CalPic.lwo and CalDate.lwo into Modeler, and save several versions with different surface names (Picture 1, Picture 2, etc.). You'll also need to create several "Calendar Month" images. This can easily be done in even the most basic paint programs.

Once you have a few versions of both Date and Picture Pages (we'll use three of each), and a few Calendar Month images, go into Layout.

Load in the following objects: CalDate.mph, CalDate1.lwo, CalDate2.lwo, CalDate3.lwo, CalPic.mph, CalPic1.lwo, CalPic2.lwo, CalPic3.lwo, and CalRing.lwo. Also add a null object to Parent all other objects to for easy movement and rotation.

First the Picture Pages: Set Dissolve to 100% for CalPic.mph and set it as the metamorph target for all three CalPic objects. Set the metamorph amount to 100 percent for CalPic1.lwo, 50 percent for

CalPic2.lwo, and 20 percent for CalPic3.lwo. It is also a good idea to move CalPic1.lwo forward on Z just slightly (-.001), and move CalPic3.lwo back on Z slightly (.001) while remembering to create key frames.

Follow the same process for the Date Pages. Dissolve CalDate.mph to 100 percent, and set it as the metamorph target for all three CalDate objects. Set the metamorph amount to 100 percent, 50 percent, and 20 percent for CalDate1, CalDate2, and CalDate3 respectively. Again, move CalDate1.lwo -.001 on Z and CalDate3.lwo .001 on Z, and create keyframes.

Now load in three of your favorite images for the calendar pictures, and the calendar month images you created. Map the picture images, one to each of the Picture Surfaces, and map the calendar month images, one to each of the CalendarDate Surfaces. Be sure to use Planar Image Map on the Z-axis, and hit Automatic Sizing. You'll also want to be sure to turn on Smoothing for all page surfaces, so your pages don't look folded.

Next, you'll want to surface the backsides of the pages. You can do this with the images already loaded. The only trick here is once you automatically size the image on the Z-axis, change the Y value in Texture Size to a negative value. This will flip the image upside down, for the backside surfaces.

Finally, give the CalRing.lwo a silver or chrome surface and try out a test render. Ta-da! (See "Ah! Much Better!" in the Color Pages.)

Final Thoughts

This, obviously, is not the only way to do this. Bones could also be used to warp our pages in Layout, rather than using morphs. Morphs gave me more predictable results, but for those of you comfortable with Bones, give it a try. The key is to subdivide your page polygons a few times to make them warpable. Also, more pages are better. After the first three pages, you can safely repeat the image maps to save RAM. Also, to continue with the *lived-in* look, be sure to have a few dates marked off, and other notes scrawled across the monthly calendar to give the impression that the calendar actually *belongs* to someone.

As mentioned before, I would be happy to receive any suggestions or feedback for this column. Until next time, here's wishing you an imperfect world!

LWP

Arnie Boedecker exists in his imperfect world in the western suburbs of Chicago. He can be reached at (708) 213-6384 or via the Internet at awb@ais.net.

Tip of the Month

Rescue Lost Surfaces

by Brian Thomas, NewTek, Inc.

When working in LightWave 5.0, clicking a surface sphere will ask if you wish to copy the attributes of the surface sample selected to the current surface you are editing. LightWave writes a temporary ".srf" file to the Temp directory that is set up in the LightWave config file in order to do this. These ".srf" files are normal surface files that can be loaded in through the Load Surface button on the Surfaces panel.

This can be handy, say, if the user ends a LightWave session but forgot to save one of the sampled settings. He or she could simply load up the .srf file located in the temp directory and then save it with a more descriptive name.

Similarly, a surface file that is loaded or saved in LightWave is really just an object file with only one defined surface. What this means is that you could actually select Load Surface and choose an object file to load. At this point LightWave will simply load the attributes of the first surface that is encountered in the object file.

Catch the Internet Wave

LightWave 3D Meets VRML

by Arnie Cachelin

The Internet is big and it's getting bigger every day. Much of the current excitement over the Internet stems from the seemingly infectious spread of the World Wide Web. Although it sounds like something from a conspiracy theory, the Web is just a bunch of files on Internet-connected computers around the world. All these files have in common is their adherence to the HTML (Hierarchical Text Markup Language) standard. HTML files contain text and images to be attractively laid out by a viewer or browser software. Most importantly, HTML files contain links to other files, such as other HTML Web pages, anywhere on the Internet. When people create HTML pages to communicate their information (or lack thereof) to the world, they include links not only to their own information, but also to other sites they have found. These links, in turn contain their own hierarchy of links, and so on. From this dynamic set of HTML files and their haphazard links to one another, the WWW has emerged.

Text-only and now text and 2D-graphics interfaces on the Internet have spawned an incredibly complex community of virtual individuals and places. Yet they are a far cry from *my* expectations of Cyberspace. Even the coolest HTML Web page doesn't match the disembodied data spaces of William Gibson's *Neuromancer* not to mention the mall of quirky avatars in Stephenson's *Snow Crash*. Adding 3D scenes to the Web is an essential step toward immersive virtual reality.

Putting the Space in CyberSpace

VRML stands for Virtual Reality Modeling Language. It is a standard for describing 3D objects and scenes via the Internet. Like HTML pages, VRML 'worlds' can contain links to remote files. Rather than using images or text for links, VRML uses 3D objects, thus the Web browser for VRML resembles a 3D animation program more closely than a DTP program. VRML worlds can be embedded in HTML pages and vice versa, and the browsers are generally closely connected. While the current v1.0 incarnation of the VRML standard limits you to pretty much static scenes floating in space, version 2.0, due to be released at SIGGRAPH '96, will add support for scripted, animated objects, collision and proximity sensing, and niceties like backdrops, fog, and 3D sound effects. Combine this with the explosive growth of the Web, and the increasing availability of low-cost 3D accelerator hardware, and it seems reasonable to believe that the 3D Web is also on the verge of explosive growth. LightWave 3D animators have the tools and skills to catch this wave

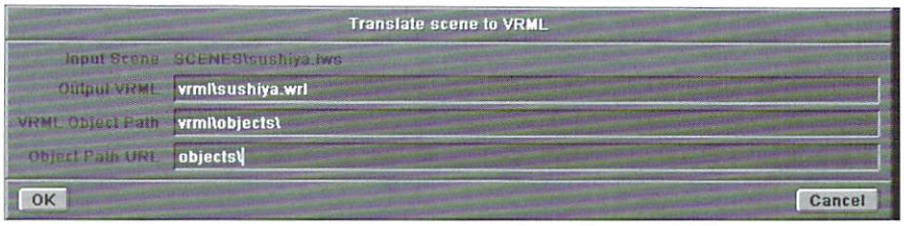


Figure 1: The panel for the VRML scene export plug-in.

Figure 2: The panel for the VRML export model plug-in.

VRML models are based either on primitives-like spheres, cubes and cones, or more likely, on sets of points and polygons. Since this is basically the approach used by LightWave's polygonal models, it isn't hard to build objects with Modeler that look pretty good when translated to VRML. More often, the problem lies in making models that don't look too good for VRML. Although the format is capable of describing complex scenes, current 3D browsers are limited by the real-time rendering capabilities of their underlying computers. Thus exquisitely crafted models with painstaking detail—suitable for those print-res closeups—will fail painfully when they enter the realm of VRML renderers (You can actually get a decent approximation of a LightWave object's VRML appearance and rendering speed from the OpenGL preview on the Windows and SGI versions of Layout or Modeler.)

To avoid the twin perils of long download times and slow rendering, remember: The first key to VRML success is efficient, low-polygon count models. Similarly, elaborate layers of diffuse, specular and luminosity textures, whether images or algorithmic, will not survive any conversion to VRML. Don't even ask about bump maps, displacement maps, or shaders. Love it or leave it, VRML supports a single image map for a color texture, as well as having diffuse color, specular and transparency values. Since that image may very well have to fly through a modem someday, you'll probably want to keep it small too.

LightWave Meets VRML

The VRML files produced by LightWave are text files that follow LightWave's style of separate object and scene files. This is not a requirement of VRML, but a powerful feature that allows a VRML scene to include objects in different files, even from some remote library. These "In-line" objects in the scene file consist of a file URL, a bounding box, and a set of position, rotation and scaling transformations.

The URL, or Universal Resource Locator, specifies a file stored in a computer somewhere in the world. It is a critical element in the formation of the World Wide Web. The bounding box information is generally used by browsers to render stand-ins while the objects are loaded.

VRML scenes also include multiple point lights, directional lights, and spotlights with adjustable cones. Although LightWave scenes currently only have one, there may be several cameras that browsers can use to jump between points of views. VRML objects created by LightWave include a set of standard viewpoints for the object.

In LightWave, VRML object output is organized along the same lines as LightWave object creation. A list of X, Y, Z coordinates describe the vertices in the object, and a list of surface names and characteristics are defined. Since these items are the easiest to modify by hand, they appear at the top of the object file.

For each surface, there is also an "IndexedFaceSet," which has the polygons with that surface. They are described as groups of numbers, referring to a vertex in the main list of point coordinates. If the original LightWave object had a color texture map image, there will be an image file name and a set of texture coordinates. Texture coordinates, also known as U, V coordinates, are 2D pixel positions in an image. They describe how the image lies on the 3D surface, by pinning certain pixels to each polygon's vertices. These values are calculated from LightWave's mapping and texture size settings. In the case of planar mapping, U and V are simply X and Y, (for Z-Axis Planar).

Spherical mapping yields U, V coordinates somewhat analogous to longitude and latitude, with the Us all bunching up at the poles. Cylindrical mapping uses Us from the spherical case, then the Vs are the coordinate lying along the texture axis.

The entire object is embedded in a VRML "anchor," which makes it an active link in the Web. If you supply a URL for the object when you create it, then any time that object appears in a scene, it will act as a clickable link to some other page. This should be used sparingly: Think about how annoying it can be to keep jumping around the Web when all you really want to do is just inspect an object.

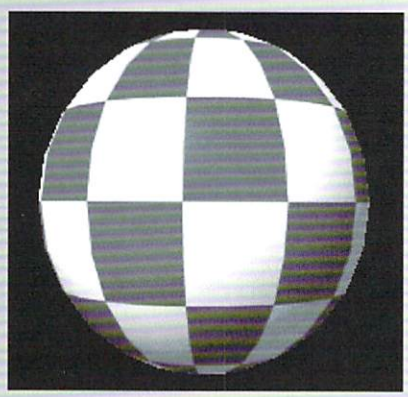
A recent addition to LightWave's VRML-saving software will embed multiple URLs in an object, tied to surfaces named such as "URL:http://www.whatever." The uses for URLs in your objects can range from billboards or ads for your favorite Website, to inventory data for some widget. A very nice example is the VRML origami site, where each step in the folding of a paper menagerie has a simple model with a link to the next stage.

This is similar to the VRML level-of-detail mode, where multiple models are grouped together and the viewer's distance determines which model, if any, is actually rendered.

Of GIFs, JPEGs and the Web

Web browsers can support a multitude of media types, either directly or through plug-ins. Despite this, two image formats are dominant on the Web: JPEG, a standard formulated by the "Joint Photographic Experts Group," and GIF, Compuserve's "Graphics Interchange Format." JPEG format (.jpg) is useful because it offers tremendous compression of 24-bit color images. The price is that the compression changes the image, throwing away some information, which can never be recovered. The amount of loss can be controlled, and many JPEG images actually throw away no data. The compression is still very good.

GIF images use a fixed palette of 2 to 256 colors, which by itself is a massively lossy form of compression. They are particularly suited for logos, graphic adornments, and buttons on Web pages because they have a transparent color, where the Web page below shows through. GIF images can also be saved in an "interlaced" format, which allows partially transmitted images to appear at full size, in lower resolution, and fill in progressively as the rest of the image is received.



Thomas Boutell and the Quest Center at Cold Spring Harbor Labs (http://www.boutell.com/gd/) has created a freely distributable software library for creating GIF images. I made it into a freely distributable image saver plug-in for LightWave 3D. Download the file gif.p from: http://www.newtek.com/cool/staff/arnie/lwgifsaver.html. Currently, only Windows Intel and Alpha versions are available.

Gif.p adds two entries to the Image Format list, Gif and Gif_NoBkg, both save 256-color interlaced images. Gif_NoBkg uses the rendered alpha channel to fill the image background with a transparent color. Because GIF uses a single color for transparency, there may be some newly visible jagged edges around your objects, which were previously smoothed over by LightWave's anti-aliasing. There are a few things you can do to minimize this problem.

When LightWave anti-aliases an object which is placed over only the background, it blends the object colors with the background colors just along the edge. Since LightWave output GIF images use LightWave's alpha channel for determining the transparent parts of an image, none of the background will show up in the image, it will be replaced with whatever background the browser of display program specifies. If you know in advance what color this is, you can make sure the visible edge pixels are blended with an appropriate color by simply setting the LightWave background to match the Web page.

Making Worlds

To create a VRML object, load some suitable object into Modeler. If it is a complex model, you might want to eliminate some polygons by removing unseen faces or poly-laden details like the steering wheel of a truck. When your model is ready, select the SaveVRML plug-in from Modeler's Objects/Custom pop-up. Select a VRML .wrl file name to save to, then modify the fields in the SaveVRML control panel as you like. The Object Name field specifies the internal VRML object name and is pretty much irrelevant. The link URL is the Web link for that object. Although this is filled in initially for (in?)convenience, you should consider making that field empty. Finally a comment is embedded in the object, if you specify one. This may be where you take credit for the model, put a less obtrusive URL, or just say No Comment. Click OK, and your .wrl is done.

If your model has a texture map image associated with it, there are a few tricks that can minimize the nuisance of hand-editing your VRML models. Because some browsers will have to load the image named in the object, it's critical to save that image name in the LightWave object. It pays to use LightWave's content directory system, so that the image path will be relative to that content directory (i.e. images\wood.jpg). It may even pay to move the image to the content dir (or vice versa) so the name in the object will have no path, and browsers will seek the image in the same directory as the object. In any event, wherever the VRML object finally resides, you will want a matching directory hierarchy in which the browser will find the image. Or you can just edit the VRML file.

Another image issue is that of image format. JPEG and GIF images are almost universally sup-

see VRML, page 18

Flight School

Part Three of Creating a 3D Bird

by James M. Curtis

Welcome back to my tutorial series on creating and animating a 3D bird. In the first two installments, we modeled and surfaced the bird object. This part details how to create the fluid simulation of flight.

Because this is an advanced tutorial, it is recommended that you have a thorough knowledge of LightWave 3D Layout. For this tutorial I'll be using version 3.5, but version 3.1 and the newer 4.0 and 5.0 can be used as well.

The screen shots and settings are based on my bird object. Since your bird may differ somewhat in size and shape than mine, you may need to make adjustments to some of the settings given here. However, the basic procedures, placement, and envelope settings should still apply.

Flight Connections

It's now time to put all the parts together. But first, because it will make moving the character around in 3D space a lot easier, we need to create some Null Objects. So, create three of them and rename them Rotation_Null, ForwardMove_Null, and Up-DownMotion_Null.

Next, load in the bird object parts in the following order. Then parent them as shown below.

WM_R-FullUp.lwo parent: Up-DownMotion_Null
WM_R-MidUp.lwo parent: none
WM_R-MidDwn.lwo parent: none
WM_R-FullDwn.lwo parent: none

WM_L-FullUp.lwo parent: Up-DownMotion_Null
WM_L-MidUp.lwo parent: none
WM_L-MidDwn.lwo parent: none
WM_L-FullDwn.lwo parent: none

Bird_Tail.lwo parent: Bird_Body.lwo

Bird_Body.lwo parent: Up-DownMotion_Null

Bird_UpLowLeg_L.lwo parent: Bird_Body.lwo

Bird_Foot-L.lwo parent: Bird_UpLowLeg_L.lwo
Bird_Head.lwo parent: Bird_Body.lwo
Bird_UpLowLeg_R.lwo parent: Bird_Body.lwo
Bird_Foot-R.lwo parent: Bird_UpLowLeg_R.lwo

Now, parent the Null Objects in this way:

ForwardMove_Null parent: Rotation_Null
Up-DownMotion_Null parent: ForwardMove_Null
Rotation_Null parent: none

Be sure everything above is correct. If it isn't, the animation we are about to set up won't work properly.

If you created your bird object parts as instructed in the last tutorial (in relative positions in Modeler), everything should line up properly as one complete whole. Figure 1 shows what my bird looks like from the Top, Side, and Front View. Yours should look similar to it.

If something doesn't line up properly, use the mouse to move the offending item(s) in the X, Y, and Z axis and create a new Key Frame (Return key) at Frame 0 as needed to correct the problem.

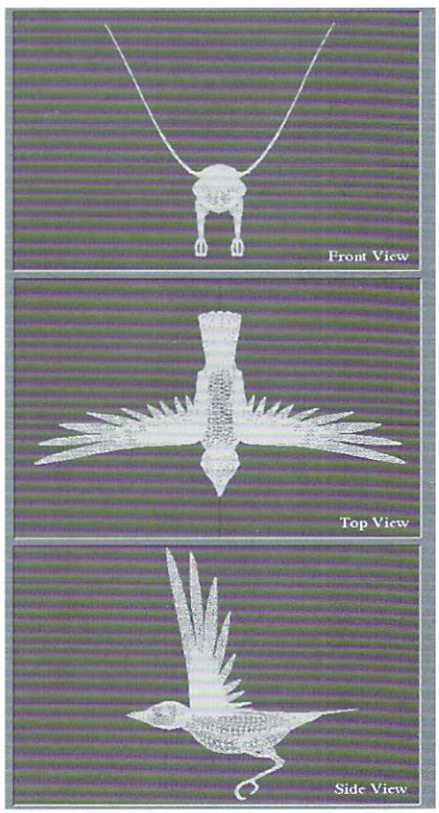
Note: To make it easier to see all of the necessary bird parts, I recommend that you set Object Dissolve (Objects panel) to 100% for the following objects:

WM_R-MidUp.lwo
WM_R-MidDwn.lwo
WM_R-FullDwn.lwo
WM_L-MidUp.lwo
WM_L-MidDwn.lwo
WM_L-FullDwn.lwo

The above six objects are the Morph Targets for the wings. As morphing targets they only affect the shape of the WM_R-FullUp.lwo and WM_L-FullUp.lwo objects. They don't need to be seen. Alternately, you could just move them out of camera view. But it is safer to dissolve them. Then they won't accidentally be rendered in the scene.

Flight Adjustments

After parenting, notice that your bird parts are still positioned where they were when first loaded. This is as it should be. However, there is a small problem. If you check through each object, you'll notice that the Pivot Point (or center of rotation) is at the 0 X, 0 Y, and 0 Z position. This must be changed for some of the objects so that when an object is rotated (Heading and Pitch most likely) it can pivot at the correct point. The way it is now, this isn't possible.



This is how my bird model appears in Modeler's Front, Top and Side views.

For example, rotate the Bird_Head.lwo part to the left or right. See how it separates from the body as it is rotated? This happens because the pivot point is in the wrong place. It should be located between the intersection of the head and neck, but it isn't. This same problem exists for some of the other objects, and is easily corrected by selecting Move Pivot Point (Main Layout Screen) for the object and moving it to it's proper location. This

normally will be at a point that connects the two objects (child to parent) and allows for natural rotational movement. Your pivot point positions may vary due to the differences in size and shape of your bird.

One other thing to check is the placement of the adjoining edges of the WM_R-FullUp.lwo and WM_L-FullUp.lwo wing objects to the Up-DownMotion_Null. Be sure that they butt up against the sides of the Bird_Body.lwo object properly toward the top edge. If not, select each wing object and move them so they join correctly.

Once finished with all these adjustments, you are ready to go to the next step... Morphing.

Winging It

To make the bird flap its wings throughout the animation, a series of morphs are used. To complete one full wing flap cycle requires the use of the four left and four right side wing positions created in the last tutorial. The wing flap cycle is created by setting up Metamorph Target Envelopes for each of the wings and repeating the motion.

Now, for each of the Current Objects below, set the Metamorph Target and corresponding Envelope accordingly.

Current Object:
WM_R-FullUp.lwo
Metamorph Target: WM_R-MidDwn.lwo
Keys
0 0%
4 100%
14 100%
20 0%
End Behavior Repeat
Spline Control Linear on all Keys
Object Dissolve 0%

Current Object:
WM_R-MidUp.lwo
Metamorph Target: WM_R-FullUp.lwo
Keys
0 0%
14 0%
20 100%
End Behavior Repeat
Spline Control Linear on all Keys
Object Dissolve 100%

Current Object:
WM_R-MidDwn.lwo
Metamorph Target: WM_R-FullDwn.lwo
Keys
0 0%
4 0%
9 100%
20 100%
End Behavior Repeat
Spline Control Linear on all Keys
Object Dissolve 100%

Current Object:
WM_R-FullDwn.lwo
Metamorph Target: WM_R-MidUp.lwo
Keys
0 0%
9 0%
14 100%
20 100%
End Behavior Repeat
Spline Control Linear on all Keys
Object Dissolve 100%

Current Object: WM_L-FullUp.lwo
Metamorph Target: WM_L-MidDwn.lwo
Keys
0 0%
4 100%
14 100%
20 0%
End Behavior Repeat
Spline Control Linear on all Keys
Object Dissolve 0%

Current Object: WM_L-MidUp.lwo
Metamorph Target: WM_L-FullUp.lwo
Keys
0 0%
14 0%
20 100%
End Behavior Repeat
Spline Control Linear on all Keys
Object Dissolve 100%

Current Object:
WM_L-MidDwn.lwo
Metamorph Target: WM_L-FullDwn.lwo
Keys
0 0%
4 0%
9 100%
20 100%
End Behavior Repeat
Spline Control Linear on all Keys
Object Dissolve 100%

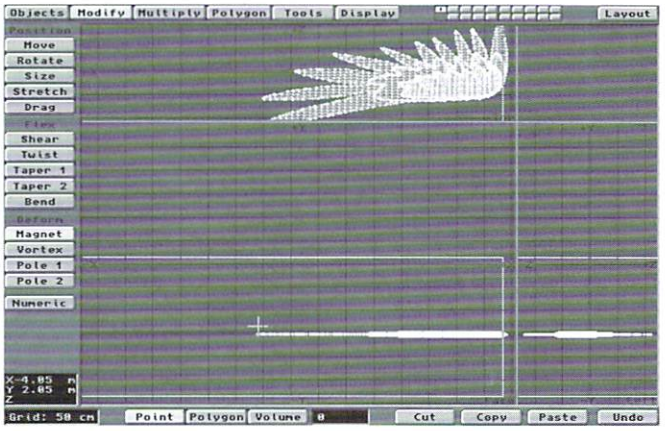


Figure 2: The mid-point of the flapping sequence.

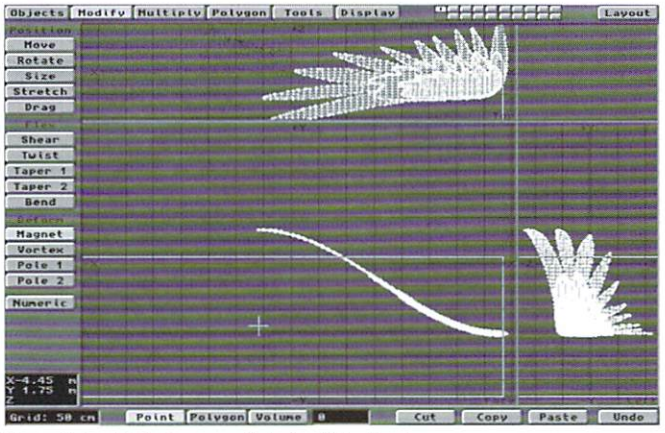


Figure 3: A point in the upward stroke of the flapping sequence.

Current Object:
WM_L-FullDwn.lwo
Metamorph Target: WM_L-MidUp.lwo
Keys
0 0%
9 0%
14 100%
20 100%
End Behavior Repeat
Spline Control Linear on all Keys
Object Dissolve 100%

You should now have all of the morph targets set up to successfully see the wings flap in a loop of 20 frames. Try it out by moving the Current Frame slider from 0 to 30 (the Scene default). For a better look at the wing flying motion, render a Preview animation of the sequence. You'll notice that you now have one complete flap motion, plus half of another. To create four complete flaps, set the Scene End Frame to 80.

If the wing flap cycle didn't work, there is an error in your settings somewhere. Re-check all the Current Object and Metamorph Targets, and then try a Preview again.

Flight Path

At this point, the bird should be flapping its wings and flying in place. Now, let's add a bit of up and down body motion to simulate the effect of the air's lift and drag on the flapping wings.

- Since the cycle is 20 frames in length, the keys should be placed at frames 0, 10, and 20.
- Move the Up-DownMotion_Null downward on the Y axis to -0.200 and create keys on frames 0 and 20. This is the point where the wings should be at their highest. Now, move the Up-DownMotion_Null upward on the Y axis to +0.200. This is where the wings should be at their lowest. Create this key at frame 10.
- Because we need the motion to continue for all flaps, go into the Motion Graph and set End Behavior to

Repeat. By adding this slight motion we have greatly increased the flight realism. If you wish, do another Preview animation to see how it looks now.

All of the basic elements needed for any bird flight animation are now set up. This is a good time to save your scene. Save it as "BirdFlight_Basic.lws".

Straight Flight

We still have another thing to cover in this tutorial—making it fly around! This is accomplished by using the ForwardMove_Null. For the following set up I recommend using the Side View (Y, Z) since it gives the easiest look at the moves being made.

•First, create a key (Return twice) on frame 0 at the current position. Go to frame 120 and move the ForwardMove_Null in the +Z direction about 20 times the length of the bird.

•Now move it upward on the +Y axis 10 to 15 times it's body height and create the key. Try to keep the angle of incline between 30 and 40 degrees.

•Go to frame 60 and move the null downward slightly to form a smooth arc. Now make a key here.

•Next, adjust the Pitch of the null on frames 60 and 120 so that the bird's body lines up parallel to the null's path, and then re-create the keys.

•Make a Preview animation to check it out.

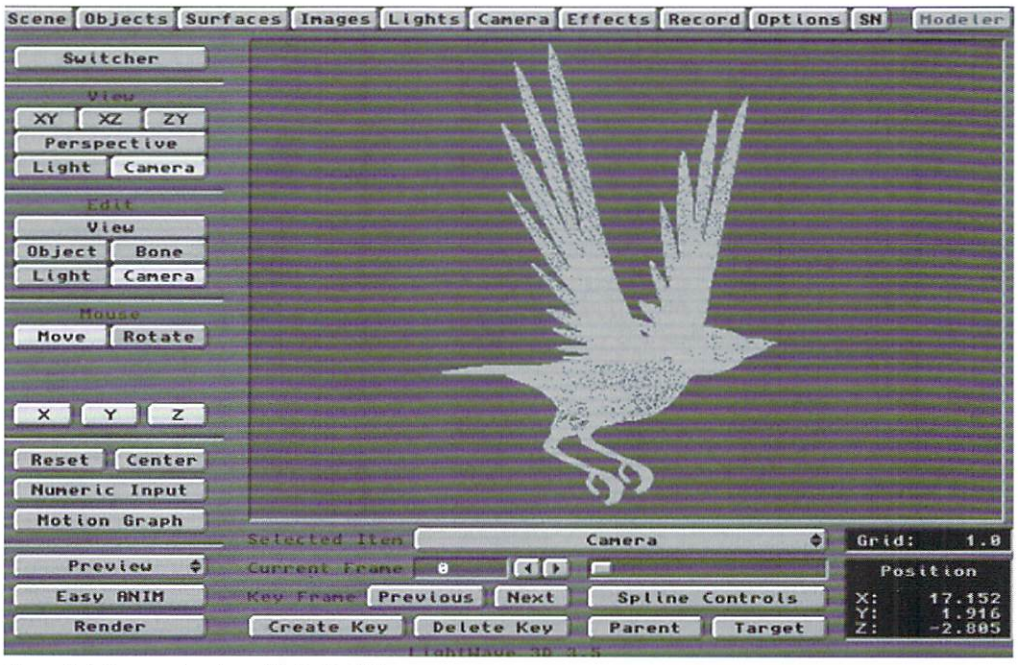


Figure 4: A Camera view of our 3D bird in flight.

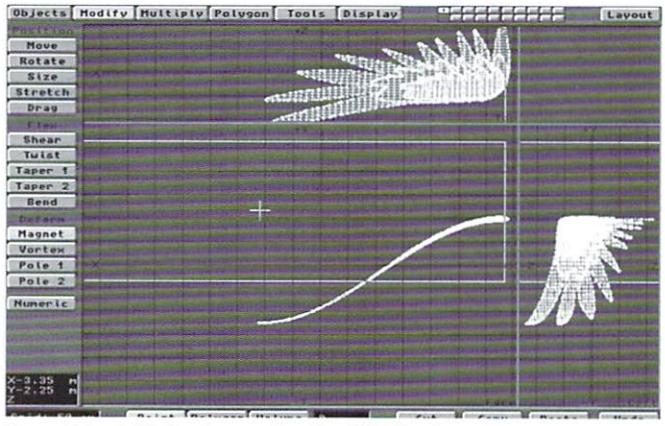


Figure 5: A point in the downward stroke of the flapping sequence.

Depending on the size of your bird object, you may need to make adjustments in the height and length of the flight path. but this should give you an idea of how it works. Save this scene as "Birdflight_Straight.lws".

Flight on a Curve

If you want to make curved flight paths, all you need to do is set the appropriate keys and adjust the ForwardMove_Null's Heading and Pitch to line up with the path. Keep in mind, as the bird changes it's Heading, it should also Bank slightly in the direction of the turn.

If you don't want to manually adjust the Heading, Pitch, and Bank for all the keys, you could just set Align To Path in the null's Motion Graph. But I find that manual setting of the rotation allows for greater flexibility in the degree of rotation and results in a more natural feeling and individual character to the bird's flight.

Flying in Circles

You may have noticed that in doing the tutorial we haven't yet used the Rotation_Null object. We'll use it now to allow the bird to fly in a completely circular path!

•To do this, Set your scene length to 180 frames.

•Delete all the ForwardMove_Null keys after frame 0.

•Then, at frame 0, move the ForwardMove_Null in the -X direction about 10 times the width of the bird. Select Rotate, and set Bank to -20.0.

•Hit Return twice to set the key. This sets the outer edge of the circular path.

•Now, while still at frame 0, select the Rotation_Null and make sure it's Heading = 0. Create a key here.

•Next, go to frame 180 and set the Numeric option to Heading = 360.0. Make the key frame.

You now have a bird that flies in a complete circle—and a perfect loop! Save this as "BirdFlight_Circle.lws".

If you want the bird to circle and climb, go to frame 180, select Move, and drag the Rotation_Null upward on the Y axis about five times the bird's height. Now select Rotate, and set the Pitch to be parallel to the incline. Hit Return twice to set the key. To continue the upward spiral for twice the time and distance, simply double the frames and values.

Flight Options

Okay, so there you have it. You have now completed some fairly realistic birdflight simulations. There are many more things that can be done. One that comes to mind is making the bird flap its wings for a while, glide, and then continue flapping its wings. This can be done by using the wing morph targets. Try to figure this one out on your own. It's really not that hard.

By the way, if anyone does create their own animations from the information in the BirdFlight tutorials, I'd love to see them (or any other animations for that matter). Send them to me on a VHS tape at the address below. Anyone who does this will get a free copy of the cool Dragon object I'm currently working on!

LWP

Jim Curtis is currently looking into the 3D video game market and hopes to bring his Vengeance Quest *characters into the games. He can be contacted by voice at (216) 354-6239.* Vengeance Quest © *1996 JMC Graphics*

Rendering for Film

continued from page 7

simple as connecting the Exabyte drive (they're SCSI) and learning TAR, the tape archive format preferred by most services. Just put all the files in a directory and back it up. On other platforms it becomes a little more complex. You'll have to get your hands on a copy of TAR that runs on your system and get a tape handler. For NT you can use Hamilton C-Shell, which has all the UNIX commands. For the Amiga you can track down a copy of BTN Tape (better than nothing, get it?) and TAR.

All the backups discussed here require a lot of typing in shells. As far as I know there is no simple "point, click and ship" software available to do this. Neither the built in back-up in NT or SGI seems to be supported by the services out there. Some companies do allow you to send 4 mm back-ups or Syquests. And some allow you to send your images in formats other than SGI. But if you want to be compatible and spare yourself extra costs, stick with the SGI frames on Exabyte. It is also possible to output different resolutions, from D1 to 4K, but 2K is a happy medium and the most widely used.

Summary

Overall, the process of rendering to film is pretty simple. The information gathered here should be enough to get you started, But there's plenty more to learn. There is nothing quite like seeing your hard work on the big screen. Be it for a major motion picture or just part of you're college's film festival, there's nothing more rewarding. I hope all of you eventually get to experience this. Now, get back to animating!

LWP

Tom Williamson is vice president/effects supervisor for Computer Café Inc. in Santa Maria, Calif. He's happily domesticated with his wife Dianna, his son Austin, his two cats, Lance and Morgan, his dog, Rascal, and hundreds of action figures. Reach him on-line at: tomcat@terminus.com

From the Net

continued from page 5

Create a key for it here at frame 0.

•Next, add a child bone by pressing the equals key (=), or by clicking Add Child Bone from the Object Skeleton panel. Rotate this bone 90 degrees on it's heading, and set the Rest Length for this bone as the width of the page. Also, move it to the top of the page. Create a key at frame 0 for this bone as well.

•Go back to bone 1, and add one more child bone. Again, rotate it 90 degrees on the heading, but this time set the bone towards the base of the page. Don't forget to create a key at frame 0 when it is in place.

•When the bones are in place, press (r) to make them active.

•To animate the page, start by rotating the first bone on its Bank. Then, hold the bottom bone (bone 3) down for a few frames while you rotate the top of the page bone (bone 2) on it's Pitch. The result is like the one in Figure 6.

That's it! You've just made a book and animated a page turn. From here, you can add more pages to turn, and even partially turned pages that remain stationary. If you'd like, add a few more bones to the right edge of the page to add a curls, and randomness to it. Place the right image map on the object, and you're all set. See the final image in the color pages of *LIGHTWAVEPRO*.

Hopefully, this opened up some new ideas for you on how to model and animate a simple everyday thing. Too often, animators think too much and over-do it, when usually there is a simpler and faster way to do things. Next month, plan on more tutorials and information "From The Net."

LWP

Dan Ablan is the president of AGA Digital Studios in Chicago, and the author of The LightWave Power Guide, *due out this September from New Riders Publishing. You can contact Dan at dma@mcs.net.*

Editor's Message

continued from page 3

Now, seaQuest has been canceled, and I am co-founder and a major partner in Digital Muse, doing what I love to do and getting paid for it!

OK, the point of all of this isn't to gloat, but to tell you that anything is possible if you put some effort into it. I was working on being a lifer for a swimming pool company and decided that I wanted to do something that I would like to do instead of something that was safe and secure, but not enjoyable anymore. Taking the initiative and finding a company that would allow me to learn from them as it helped better my position was all it took to get me out to L.A. Now I'm co-owner of a successful company and I'm making television and movies!

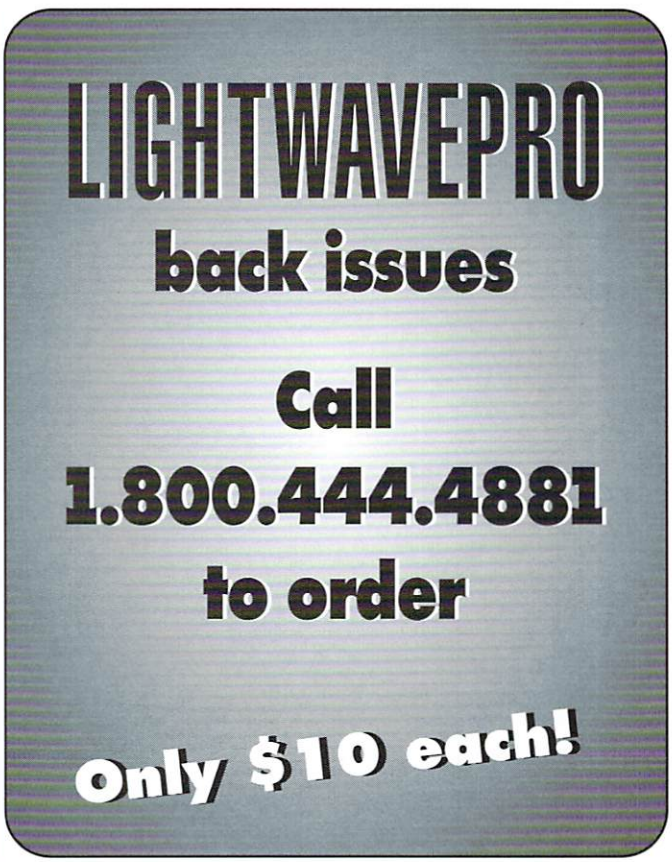
If your work isn't what you love, then something isn't right. (a David Byrne quote for those of you that may not have caught it) Make it what you love by taking a chance and doing something about it. I really believe that you should do what you love and money will follow.

On the NewTek Front

During SIGGRAPH, NewTek will be announcing a film grain plug-in created by Steve Worley of Worley Labs. It will be distributed by NewTek for about $50 and it is definitely worth the price. I've used it in testing, and it works rather well. It allows you to grain your LightWave renderings or even de-grain a film plate. You'll want to remove grain from a captured film plate in order to add your LW graphics to it and then re-grain over the whole thing.

Here's hoping that you are doing what you love (at least sometimes)!

-John Gross,
Editor



VMRL

continued from page 13

ported on the Web. Although LightWave only loads and saves JPEG images, there is a free plug-in to save Web-friendly GIF images available (See sidebar). JPEG images are nice and small. While their high-powered yet lossy compression causes visible artifacts, they should be virtually invisible at Web/VRML resolutions. If you have nice high-quality texture images for your rendering work, and want VRML versions, make smaller, JPEG versions of the images for the Web. A happy legacy of LightWave's (and HIIP's) non-PC origins is that file name extensions don't really matter. So, although it may cause some confusion, you can rename your texture.tga to texture.jpg, and use that on your model. When you install the VRML model, just use the smaller JPEG image. Or you can just edit the VRML file.

Make the Scene

Making a scene is not much harder than making an object. Because VRML scenes need their objects' bounding boxes, the Scene2VRML plug-in runs in Modeler (where it can load each object in the scene) quickly get its measurements and moves on. As a result, it can also determine whether the VRML version of the object exists yet, and run SaveVRML for you *automagically*. If the VRML object exists already, Scene2VRML plays it safe and leaves it undisturbed.

Once you have a LightWave scene to make into VRML, select Scene2VRML from Modeler's Objects/Custom pop-up. Select the scene in the file requester, then set the fields in the Scene2VRML control panel. Output VRML is the VRML scene to save, by default, the same as the LightWave scene with a .wrl extension. The VRML Object Path field should have the directory path where the VRML versions of your objects are stored. This is where Scene2VRML will look for the objects, and will create them if they don't already exist.

This is half the confusing part. The other half is that the Object Path URL should be the same directory, but formatted as an URL, for the benefit of the browser loading the VRML scene. So on your machine, the .wrl object files may reside in "C:\webpage\vrml\objects\", while the URL for it might be "FILE:///C|/webpage/vrml/objects." If the scene lived in "C:\webpage\vrml\," then you could get away with specifying just objects/ for the URL. Similarly, if the scenes and objects are in the same place, the URL could be empty. Because the browser finds the object through a full-blown URL, rather than just a filename (as images currently do), scenes all over the Internet could refer to the same object. So if you want to use a remote object, or are eventually setting the VRML scene up on a different computer, specify the correct URL. It doesn't alter Scene2VRML's behavior. Or you can just edit the VRML file.

The Future and VRML

Version 2.0 of VRML will significantly extend the capabilities of VRML worlds not only by making the worlds prettier (or at least, closer to your LightWave scene), but by adding powerful interactivity, animation and scripting. Even now, NewTek's VRML team is working to bring these new features to you. Things that are natural for LightWave, like non-linear fog and a backdrop image will translate to VRML 2.0 scenes. In fact, VRML 2.0 supports six backdrop images for front, back, top, bottom, left and right views. Objects will have keyframed motions, so lights and even colors can be enveloped. And there also will be ways of doing things that LightWave 3D does not directly support. These will likely include collision detection, visibility/proximity/touch sensing, and level-of-detail objects.

VRML2.0 has some nice texturing improvements, including a 2D transformation on an image or movie(!) texture map, and a list of URLs in place of an image file name. If you need more info, check out www.vag.org (the site for the VRML Architecture Group).

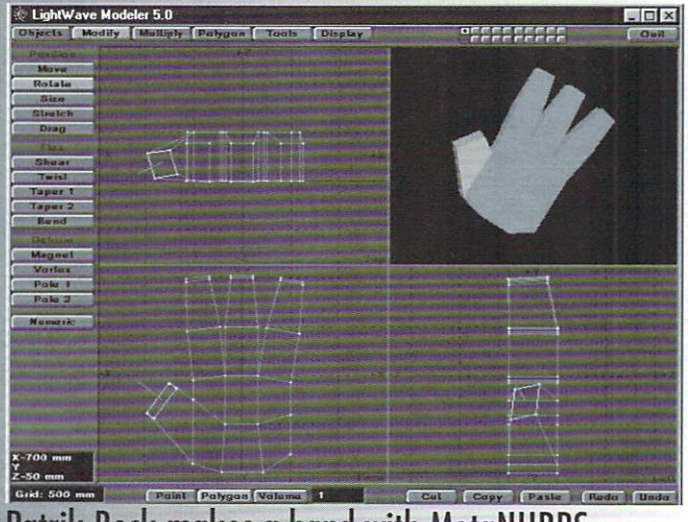
VRML is maturing into a comprehensive standard for creating and sharing virtual realities that will broaden the appeal and extent of 3D cyberspace. There's infinite room for virtual models, so why not populate this brave new world with some of your own creatures?

LWP

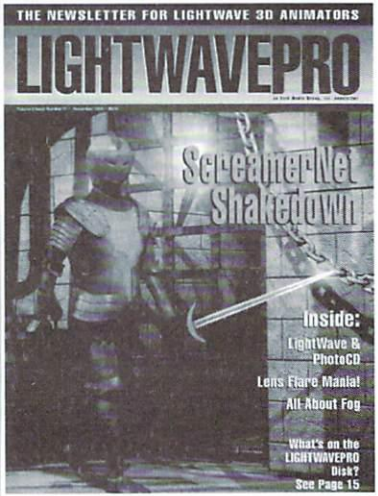
COMING UP NEXT IN

LIGHTWAVEPRO

- More treasures in Dan Ablan's 'From the Net'
- Arnie Boedecker's journeys through an 'Imperfect World'
- David Ebner explores LightWave new surfacing capabilities



Patrik Beck makes a hand with MetaNURBS.



LIGHTWAVEPRO

THE NEWSLETTER FOR LIGHTWAVE 3D® ANIMATORS

Make a Smart Investment

Subscribe Today!

Canadian/Mexico: Add $US12
Overseas: Add $US36
Allow 6-8 weeks for delivery.

Make checks payable to LIGHTWAVEPRO.
Prepayment required on all overseas orders.

YES! Enter my one-year (12 issues) subscription to LIGHTWAVEPRO at the Special Introductory Rate of only $48—that's 50% off the cover price!

Name_____

Address_____ Apt. #_____

City_____ State _____ Zip _____

☐ Payment Enclosed ☐ Bill Me

46-GLO-LWP

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

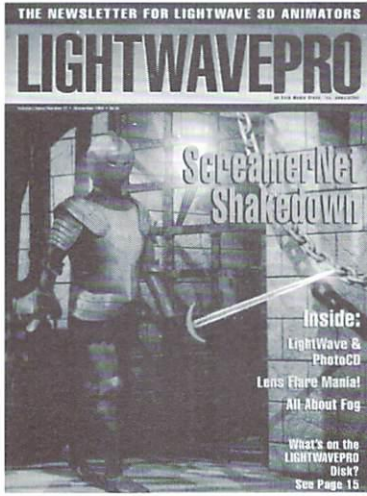
BUSINESS REPLY MAIL

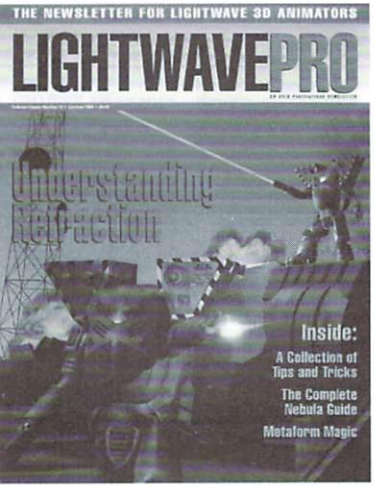
FIRST-CLASS MAIL PERMIT NO. 1286 BOULDER, CO

POSTAGE WILL BE PAID BY THE ADDRESSEE

LIGHTWAVEPRO

P.O. Box 54240
Boulder, CO 80323-4240





LIGHTWAVEPRO

THE NEWSLETTER FOR LIGHTWAVE 3D® ANIMATORS

Make a Smart Investment

Subscribe Today!

Canadian/Mexico: Add $US12
Overseas: Add $US36
Allow 6-8 weeks for delivery.

Make checks payable to LIGHTWAVEPRO.
Prepayment required on all overseas orders.

YES! Enter my one-year (12 issues) subscription to LIGHTWAVEPRO at the Special Introductory Rate of only $48—that's 50% off the cover price!

Name _____

Address _____ Apt. # _____

City _____ State _____ Zip _____

☐ Payment Enclosed ☐ Bill Me

46-GLO-LWP

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

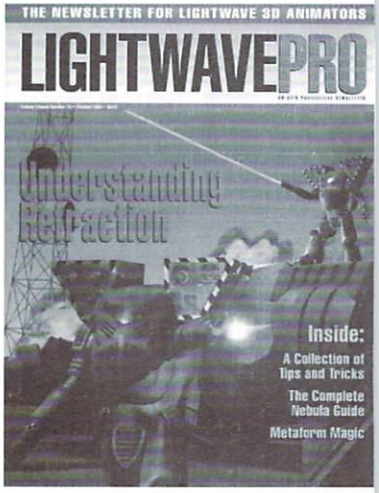
BUSINESS REPLY MAIL

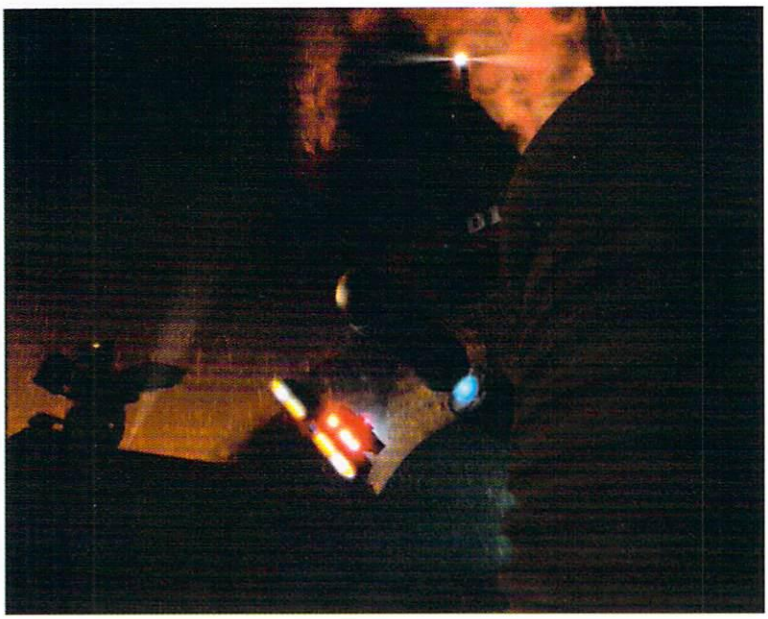
FIRST-CLASS MAIL PERMIT NO. 1286 BOULDER, CO

POSTAGE WILL BE PAID BY THE ADDRESSEE

LIGHTWAVEPRO

P.O. Box 54240
Boulder, CO 80323-4240





Raising Havoc

More images from Dinko Cepak's four-minute music video. At left, a plume of smoke rises above the battlefield.
Copyright©1996 Dinko Cepak

On the Run

Cepak's mechanoid-like vehicle strides under the arches of a bridge.
Copyright©1996 Dinko Cepak

Vengeance Quest

Some of the characters created by animator James M. Curtis and featured in *Vengeance Quest*, his 3D animated fantasy short.
Copyright©1996 JMC Graphics

LIGHTWAVE 3D™

Turn on your TV.

Star Trek: Voyager

seaQuest DSV

Babylon 5

Sliders

Hercules: The Legendary Journeys

Unsolved Mysteries

The Tonight Show with Jay Leno

RoboCop: The Series

VR5

Weird Science

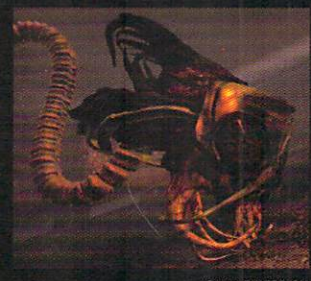
Star Trek: The Next Generation

©1995 AMBLIN IMAGING

MCA/UNIVERSAL

©1995 MCA/UNIVERSAL

©1995 AMBLIN IMAGING



©1995 CINESTASIA

The professional standard in 3D Graphics for Windows, Windows NT, Silicon Graphics Inc., DEC Alpha, MIPS, and Amiga Platforms.

FOR MORE INFORMATION CALL
1-800-847-6111

NEWTEK INCORPORATED

Outside the U.S.A. call 1-913-228-8000
FAX 1-913-228-8099
WEB PAGE HTTP://WWW.NEWTEK.COM
ANONYMOUS FTP SITE FTP.NEWTEK.COM

LightWave 3D is a trademark of NewTek, Inc. Amiga is a trademark of Commodore. DEC Alpha is a trademark of Digital Equipment Corp. MIPS is a trademark of Silicon Graphics, Inc. Windows, Windows NT are trademarks of MicroSoft, Inc. © NewTek, Inc. 1995. *Design, FryeAllen, Inc.*